CLAIMS

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is as follows:

1. A method of multicast transmission on a network processor comprising the steps of:

storing the frame to be transmitted in a series of buffers, chained together by a linked list;

associating a buffer control block with each buffer;

associating a frame control block with each frame;

receiving frames into a queue to await dispatch by a network processor;

associating a queue control block with the queue of frames to be transmitted;

assigning additional buffers and additional frame control blocks for each multicast target and linking these additional frame control blocks with the original frame control block associated with the frame;

using a multicast counter to determine when the frame has been sent to each of the multicast targets;

returning the buffers and frame control blocks assigned to each multicast target to free queues as the frame is sent to each target; and

returning the original buffers and frame control blocks to the free queues after the frame has been sent to all of the multicast targets.

2. The method for multicast transmission as recited in claim 1, wherein the buffer control block associated with each buffer forms a linked list for

chaining buffers into a frame and contains a plurality of fields, including separate fields to

store a pointer to the next buffer in the frame;

store the offset of the first valid byte of data in the next buffer of a frame;

store the offset of the last valid byte of data in the next buffer of a frame; and

indicate whether the next buffer in the frame should be returned to the free buffer or queue or retained so as to continue multicast transmission.

3. The method for multicast transmission as recited in claim 1, wherein the frame control block associated with each frame forms a linked list for chaining frames into a queue and contains a plurality of fields, including separate fields to

store a pointer to the next frame in the queue;

store a count of the total number of bytes of the next frame in the queue;

store the address of the first buffer in a frame;

store the starting byte position of valid data in the first buffer of a frame;

store the ending byte position of valid data in the first buffer of a frame; and

store information on the format and the type of the frame to be transmitted.

4. The method for multicast transmission as recited in claim 1, wherein the step of receiving frames into a queue comprises the further steps of:

popping a free buffer address from the head of the free buffer queue;

popping a free frame control block from the head of the free frame control block queue;

writing frame data to the buffer;

writing control information, including the first buffer address, the starting and ending byte positions for valid data in the first buffer, to the frame control block;

setting a working byte count register to the number of bytes written to the first buffer;

repeating this process until the entire frame is written to buffers; and

adding the frame to the tail of an input queue to await dispatch to the network processor.

5. The method for multicast transmission as recited in claim 1, wherein the queue control block associated with the queue of frames to be transmitted includes a plurality of fields, including separate fields to

store the address of the frame control block associated with the frame at the head of the queue;

store a count of the total number of valid bytes in the frame at the top of the queue; and

store the address of the frame control block associated with the frame at the tail of the queue.

6. The method of multicast transmission as recited in claim 1, wherein static frames may be transmitted comprising the step of sending a frame to each multicast target without using the multicast counter or returning any frame control blocks or buffers to the free queues.

7. A network processor supporting multicast transmission comprising:

means for storing a frame to be transmitted in a series of buffers, chained together by a linked list;

means for associating a buffer control block with each buffer and associating a frame control block with each frame;

means for receiving frames into a queue to await dispatch;

means for associating a queue control block with the queue of frames to be transmitted;

means for assigning additional buffers and additional frame control blocks for each multicast target and linking these additional frame control blocks with the original frame control block associated with the frame;

means using a multicast counter for determining when the frame has been sent to each of the multicast targets; and

means returning the buffers and frame control blocks assigned to each multicast target to free queues as the frame is sent to each target and returning the original buffers and frame control blocks to the free queues after the frame has been sent to all of the multicast targets.

8. The network processor as recited in claim 7, wherein the buffer control block associated with each buffer forms a linked list for chaining buffers into a frame and contains a plurality of fields, including separate fields to

store a pointer to the next buffer in the frame;

store the offset of the first valid byte of data in the next buffer of a frame;

store the offset of the last valid byte of data in the next buffer of a frame; and

indicate whether the next buffer in the frame should be returned to the free buffer or queue or retained so as to continue multicast transmission.

9. The network processor as recited in claim 7, wherein the frame control block associated with each frame forms a linked list for chaining frames into a queue and contains a plurality of fields, including separate fields to

store a pointer to the next frame in the queue;

store a count of the total number of bytes of the next frame in the queue;

store the address of the first buffer in a frame;

store the starting byte position of valid data in the first buffer of a frame;

store the ending byte position of valid data in the first buffer of a frame; and

store information on the format and the type of the frame to be transmitted.

10. The network processor as recited in claim 7, wherein the means for receiving frames into a queue comprises:

means for popping a free buffer address from the head of the free buffer queue;

means for popping a free frame control block from the head of the free frame control block queue;

means for writing frame data to the buffer;

means for writing control information, including the first buffer address, the starting and ending byte positions for valid data in the first buffer, to the frame control block;

means for setting a working byte count register to the number of bytes written to the first buffer; and

means, in response to the entire frame is written to buffers, for adding the frame to the tail of an input queue to await dispatch to the network processor.

11. The network processor as recited in claim 7, wherein the queue control block associated with the queue of frames to be transmitted includes a plurality of fields, including separate fields to

store the address of the frame control block associated with the frame at the head of the queue;

store a count of the total number of valid bytes in the frame at the top of the queue; and

store the address of the frame control block associated with the frame at the tail of the queue.

12. The network processor as recited in claim 7, wherein static frames may be transmitted comprising the step of sending a frame to each multicast target without using the multicast counter or returning any frame control blocks or buffers to the free queues.